Amendment to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (currently amended): A method for performing route calculations in a link state routing protocol at a node within a computer network, the method comprising:

receiving new route information at the node;

evaluating existing routes of the node before recalculating and modifying routes to determine if said new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost;

recalculating routes and modifying a routing table for the node only when said new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost; and

upon losing one of the existing routes:

initializing a best cost;

finding a neighbor node's link information by traversing an LSP;

calculating a neighbor cost of reaching another node via the neighbor node; and

setting the best cost to the neighbor cost if the neighbor cost is less than the best cost;

wherein recalculating routes comprises modifying information about links within the network without examining each of the nodes within the network.

Claim 2 (original): The method of claim 1 further comprising receiving a link state packet with information about the node's path to a root node and wherein

the node's route to the root node is improved and further comprising evaluating the node's neighbor nodes.

Claim 3 (original): The method of claim 1 further comprising receiving a link state packet with information about the node's path to a root node and wherein the node's route to the root node has worsened and further comprising evaluating the node's path to the root node.

Claim 4 (original): The method of claim 3 wherein nodes contained within a subtree containing the node are scrapped and the routes to all nodes in the subtree are re-evaluated.

Claim 5 (original): The method of claim 1 wherein recalculating existing routes comprises implementing equal-cost path splitting.

Claim 6 (original): The method of claim 5 wherein the new route information improves existing routes and only a parent node sending the new route information is used in recalculating routes.

Claim 7 (original): The method of claim 5 wherein the new route information worsens existing routes and a parent node sending the information is no longer considered a parent node by said node.

Claim 8 (original): The method of claim 1 wherein the computer network comprises greater than one hundred nodes.

Claim 9 (original): The method of claim 1 wherein said node has lost its path to another node within the computer network.

Claim 10 (original): The method of claim 9 further comprising reattaching the node at a location within a remaining portion of a spanning tree.

Claim 11 (original): The method of claim 11 further comprising recalculating routes to all other nodes in a subtree of which the node is a root node.

Claim 12 (original): The method of claim 1 further comprising performing an incremental route recalculation for all nodes within the network that have received new link state information.

Claims 13 -14 (canceled).

Claim 15 (currently amended): A computer program product for performing route calculations in a link state routing protocol at a node within a computer network, comprising:

code that evaluates existing routes of the node when new route information is received to determine if said new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost;

code that recalculates routes and modifies a routing table for said node only when said new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost;

code that, upon losing one of the existing routes:

initializes a best cost;

finds a neighbor node's link information by traversing an LSP;

calculates a neighbor cost of reaching another node via the neighbor node; and

sets the best cost to the neighbor cost if the neighbor cost is less than the best cost; and

a computer-readable storage medium for storing the codes;

wherein said code that recalculates routes comprises code that modifies information about links within the network without examining each of the nodes within the network.

Claim 16 (original): The computer program product of claim 15 wherein the computer-readable medium is selected from the group consisting of CD-ROM, floppy disk, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

Claim 17 (original): The computer program product of claim 15 further comprising code that performs equal-cost path splitting.

Claim 18 (canceled).

Claim 19 (currently amended): A system for performing route calculations in a link state routing protocol at a node within a computer network, the system comprising a processor operable to evaluate existing routes of the node when new route information is received to determine if said new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost, recalculate routes and modify a routing table for said node only when said new route information improves existing routes or existing routes are made worse or lost, and upon losing one of the existing routes initializes a best cost, finds a neighbor node's link information by traversing an LSP, calculates a neighbor cost of reaching another node via the neighbor node, and sets the best cost to the neighbor cost if the neighbor cost is less than the best cost; wherein recalculating routes comprises modifying information about links within the network without examining each of the nodes within the network; and memory for storing route information.

Claim 20 (currently amended): A system for performing route calculations in a link state routing protocol at a node within a computer network, comprising:

means for evaluating existing routes of the node when new route information is received to determine if said new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost;

means for recalculating routes and modifying a routing table for said node only when said new route information improves existing routes or existing routes are made worse or lost;

means for initializing a best cost when one of the existing routes is lost;

means for finding a neighbor node's link information by traversing an LSP;

means for calculating a neighbor cost of reaching ~~a~~ another node via the neighbor node;

means for setting the best cost to the neighbor cost if the neighbor cost is less than the best cost; and

memory for storing route information;

wherein recalculating routes comprises modifying information about links within the network without examining each of the nodes within the network.

Claim 21 (original): The system of claim 20 further comprising means for performing equal-cost path splitting.

Claim 22 (previously presented): A method for performing route calculations in a link state routing protocol at a root node within a computer network, the method comprising:

receiving new route information at the root node;

sorting nodes with new route information into order of cost;

evaluating changes in state;

evaluating routes if existing routes are improved, lost, or made worse;

reattaching routes at lowest cost point in a spanning tree; and

re-evaluating routes from reattached nodes; and

upon losing one of the existing routes:

initializing a best cost;

finding a neighbor node's link information by traversing an LSP;

calculating a neighbor cost of reaching a node via the neighbor node; and

setting the best cost to the neighbor cost if the neighbor cost is less than the best cost.

Claim 23 (previously presented): The method of claim 22 further comprising splitting traffic across more than one path if total cost is the same for each of the paths.

Claim 24 (previously presented): The method of claim 23 wherein evaluating changes in state comprises performing incremental route recalculation.

Claim 25 (previously presented): The method of claim 22 wherein sorting nodes comprises sorting nodes into order of cost from the root node.

Claim 26 (previously presented): The method of claim 22 further comprising leaving routes unchanged if the new route information has no effect on existing routes.

Claim 27 (previously presented): The method of claim 22 wherein existing routes are lost or made worse and further comprising re-evaluating a subtree of the root node.

Claim 28 (previously presented): The method of claim 27 further comprising splitting paths among equal cost routes.

Claim 29 (previously presented): The method of claim 1 wherein each node within the computer network is represented by a data structure comprising information about links to other nodes and cumulative cost of all links traversed from root to the node.

Claim 30 (previously presented): The method of claim 1 wherein recalculating routes from the node comprises applying an incremental Dijkstra's algorithm to the node.

Claim 31 (canceled).

Claim 32 (previously presented): The method of claim 1 wherein said at least one of the existing routes is made worse and further comprising recalculating routes to all nodes in a subtree of the node.

Claim 33 (previously presented): The method of claim 1 wherein recalculating routes comprises recalculating routes at all nodes which have received new link state information and processing said nodes in increasing order of distance from a root node.

Claim 34 (canceled).

Claim 35 (previously presented): The method of claim 1 further comprising applying an incremental Dijkstra's algorithm to the root node only if said new route information improves or worsens at least one of the existing routes or at least one of the existing routes is lost.

Claim 36 (previously presented): The method of claim 35 further comprising applying equal-cost path splitting.

Claim 37 (previously presented): The method of claim 35 wherein the number of nodes examined is proportional to the log of the number of nodes within the network.